

[Home](#)[Arduino Projects](#)[Project#00](#)[Arduino +](#)[Proteus +](#)[Fritzing](#)[Project#01](#)[Cylon's Eye](#)[Project#02](#)[External](#)[Interrupt](#)[Project#03](#)[Photoresistor](#)[Input](#)**Project#04**
Solar Tracker[Project#05](#)[Using LM35 /](#)
[LM34](#)[Project#06](#)[Using I2C 8-bit](#)[IO Expander](#)[PCF8574 /](#)[MCP23008](#)[Project#07](#)[Using DS18B20 /](#)[DS18S20](#)[Project#08](#)[Using DHT22](#)[Project#09 US-](#)[100 Ultrasonic](#)[sensor](#)[Project#10](#)[Sharp GP2D02](#)[Infrared](#)[Ranging Sensor](#)[Project#11](#)[Infrared speed](#)[sensor module](#)[\(applied on DIY](#)[anemometer\)](#)[Project#12](#)[Demonstrates](#)[use of ADXL345](#)[Project#13](#)[Wireless Wii](#)[Nunchuck](#)[control on Pan](#)[& Tilt](#)[Project#14](#)[Fusion IMU3000](#)[with ADXL345](#)[Project#15](#)[Experiment on](#)[Gearmotor with](#)[Encoder](#)[Project#16 DHT-](#)[22 and HC-06](#)[Bluetooth](#)[Project#17](#)[Sonic Radar](#)[Using US-100](#)[Project#18](#)[OLED Compass](#)[LSM303DLH](#)[\(applied on DIY](#)[wind Vane\)](#)[Project#19](#)[Using MAX9812](#)[Microphone](#)[Amplifier](#)[measure dB, dB-](#)[Home](#) > [Arduino Projects](#) >

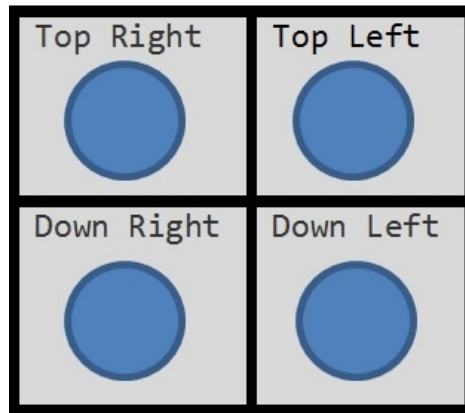
Project#04 Solar Tracker

Tracking sunlight using 4 photo-resistors and 2 servos.

Origination of the 4 photo-resistors:

Using cardboard, makes a long 3D cross with 4 sections. Place it in the center to separate 4 photo-resistors.

Attach the whole unit to the vertical servo.



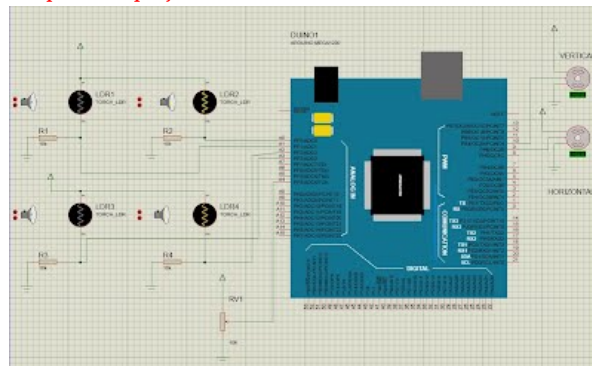
I use [LynxmotionPan and Tilt Kit](#) for 2D motion



Circuit:

Notice the connection of the photo-resistor. The brightest light on the LDR, higher value is read from the analog pin.

See [previous project](#).



[SPL, OLED VU Meter](#)
[Project#20](#)
[OLED Spectrum Analyzer using Fixed-point FFT; FHT on free-running mode](#)
[Project#21](#)
[Create own PCB shield](#)
[chipKIT Tutorials and Projects](#)
[IMU List \(last update: 06/16/2019\)](#)
[Links](#)
[Math](#)
[Microcontroller](#)
[Physics](#)
[PIC16F877A](#)
[Projects](#)
[Raspberry Pi2/Pi3](#)
[Projects](#)
[Sitemap](#)

Code:

```

1 // Reference from:
2 // http://www.instructables.com/id/Arduino-Solar-Tracker/?ALLSTEPS
3 // http://fritzing.org/projects/arduino-2-dof-sun-tracker
4 //
5 // Pins setting:
6 // pin D9 = horizontal servo
7 // pin D10 = vertical servo
8 // pin A0 = Top Left photoresistor
9 // pin A1 = Top Right photoresistor
10 // pin A2 = Down Left photoresistor
11 // pin A3 = Down Right photoresistor
12
13 #include <Servo.h>
14
15 //create servo objects to control servos
16 Servo horizontalServo;
17 Servo verticalServo;
18 int val_Vertical=90;      // set initial value of vertical servo at the middle po
19 int val_Horizontal=90;    // set initial value of horizontal servo at the middle
20
21 // LDR pin connections
22 // name = analogpin;
23 const int LDR_TL = 0; //LDR top left A0
24 const int LDR_TR = 1; //LDR top right A1
25 const int LDR_DL = 2; //LDR down left A2
26 const int LDR_DR = 3; //LDR down right A3
27 int avg_Top = 0;      // average value top
28 int avg_Down = 0;     // average value down
29 int avg_Left = 0;     // average value left
30 int avg_Right = 0;    // average value right
31
32 // Use Potentiometer 10k (value doesn't matter) for Tolerance Settings
33 const int TolerancePin = 4; // pin A4
34 int Tolerance = 1;
35
36 void setup(){
37     // servo connections
38     horizontalServo.attach(9);
39     verticalServo.attach(10);
40     // serial output
41     Serial.begin(9600);
42 }
43
44 void loop(){
45     read_potentiometer();
46     read_LDRs();
47     adjust_Servos();
48     delay(15);
49 }
50
51 void read_potentiometer(){
52     // Use Potentiometer for Tolerance Settings
53     Tolerance = analogRead(TolerancePin)/4;
54     Serial.print("Tolerance: ");
55     Serial.println(Tolerance);
56 }
57
58 void read_LDRs(){
59     // scale down 10-bit analogRead to the angle of the servo in degrees, 0 to 180
60     int val_TL = map(analogRead(LDR_TL),0,1023,0,179);
61     int val_TR = map(analogRead(LDR_TR),0,1023,0,179);
62     int val_DL = map(analogRead(LDR_DL),0,1023,0,179);
63     int val_DR = map(analogRead(LDR_DR),0,1023,0,179);
64     avg_Top= (val_TL + val_TR) / 2;      // average value top
65     avg_Down= (val_DL + val_DR) / 2;     // average value down
66     avg_Left = (val_TL + val_DL) / 2;    // average value left
67     avg_Right = (val_TR + val_DR) / 2;   // average value right
68     Serial.print("LDR top left: ");
69     Serial.println(val_TL);
70     Serial.print("LDR top right: ");
71     Serial.println(val_TR);
72     Serial.print("LDR down left: ");
73     Serial.println(val_DL);
74     Serial.print("LDR down right: ");
75     Serial.println(val_DR);
76 } //end read_LDRs
77
78 void adjust_Servos(){
79

```

```

80 int dvert = avg_Top - avg_Down;    // check the difference of up and down
81 int dhoriz = avg_Left - avg_Right; // check the difference of left and right
82
83 // Using the tolerance value stop servo seeking
84 // Check Vertical check if the difference is in the tolerance else change vertical
85 if (-1*Tolerance > dvert || dvert > Tolerance){
86     if (avg_Top > avg_Down){
87         //adjust Vertical Servo Down
88
89         val_Vertical = ++val_Vertical;
90         if (val_Vertical > 179){
91             val_Vertical = 179;
92         }
93     }
94     else if (avg_Top < avg_Down){
95         //adjust Vertical Servo Up
96         val_Vertical = --val_Vertical;
97
98         if (val_Vertical < 1){
99             val_Vertical = 1;
100        }
101    }
102    verticalServo.write(val_Vertical);
103 }
104 //Check Horizontal check if the difference is in the tolerance else change horizontal
105 if (-1*Tolerance > dhoriz || dhoriz > Tolerance){
106     if (avg_Left > avg_Right){
107         //adjust Horizontal Servo Left
108         val_Horizontal = --val_Horizontal;
109
110         if (val_Horizontal < 1){
111             val_Horizontal = 1;
112         }
113     }
114     else if (avg_Left < avg_Right){
115         //adjust Horizontal Servo Right
116         val_Horizontal = ++val_Horizontal;
117
118         if (val_Horizontal > 179){
119             val_Horizontal = 179;
120         }
121     }
122     else if (avg_Left == avg_Right){
123         // nothing
124     }
125     horizontalServo.write(val_Horizontal);
126 }
127
128 Serial.print("Servo Horizontal: ");
129 Serial.println(val_Horizontal);
130 Serial.print("Servo Vertical: ");
131 Serial.println(val_Vertical);
132 } // end adjust servos

```